

The Undecidability of Arbitrary Arrow Update Logic

Hans van Ditmarsch Wiebe van der Hoek
Louwe B. Kuijer

Abstract

Arbitrary Arrow Update Logic is a dynamic modal logic that uses an arbitrary arrow update modality to quantify over all arrow updates. Some properties of this logic have already been established, but until now it remained an open question whether the logic’s satisfiability problem is decidable. Here, we show that the satisfiability problem of Arbitrary Arrow Update Logic is co-RE hard, and therefore undecidable, by a reduction of the tiling problem.

1 Introduction

In the field of Dynamic Epistemic Logic, various kinds of *updates* are used to model events that change the information state of agents. These kinds of updates include public announcements [11, 4], action models [4] and arrow updates [10], among others. Somewhat recently, there has been a trend to enrich logics with “arbitrary” versions of such updates. See for example Arbitrary Public Announcement Logic (APAL) [3], Group Announcement Logic (GAL) [1], Arbitrary Action Model Logic (AAML) [9], and Arbitrary Arrow Update Logic (AAUL) [6, 7]. The intuition behind these “arbitrary” operators is that they represent universal quantification over their non-arbitrary counterpart; for example, φ is true after an arbitrary public announcement if φ is true after every public announcement.

For some of these logics, namely APAL and GAL, the satisfiability problem is undecidable [8, 2]. The satisfiability problem of AAML, on the other hand, is decidable [9]. For AAUL, it remained unknown whether the satisfiability problem is decidable. Here, we show that AAUL’s satisfiability problem can encode the tiling problem [13]. Because the tiling problem is known to be co-RE complete [5], this shows that the satisfiability problem of AAUL is co-RE hard.

The structure of this paper is as follows. First, in Section 2 we introduce the syntax and semantics of AAUL. Then, in Section 3 we provide a brief definition of the tiling problem and show that it can be encoded in the satisfiability problem of AAUL.

2 AAUL Syntax and Semantics

Let \mathcal{P} be a countable set of propositional variables and \mathcal{A} a finite set of agents. We assume that $|\mathcal{A}| \geq 6$.

Definition 1. The language \mathcal{L}_{AAUL} of AAUL is given by the following normal forms:

$$\begin{aligned}\varphi &::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box_a\varphi \mid [U]\varphi \mid [\uparrow]\varphi \\ U &::= (\varphi, a, \varphi) \mid U, (\varphi, a, \varphi)\end{aligned}$$

where $p \in \mathcal{P}$ and $a \in \mathcal{A}$. The language \mathcal{L}_{AUL} is the fragment of \mathcal{L}_{AAUL} that does not contain $[\uparrow]$.

We use $\vee, \rightarrow, \leftrightarrow, \Diamond, \langle U \rangle, \langle \uparrow \rangle, \bigvee$ and \bigwedge in the usual way as abbreviations. Furthermore, we slightly abuse notation by identifying the list $U = (\varphi_1, a_1, \psi_1), \dots, (\varphi_k, a_k, \psi_k)$ with the set $U = \{(\varphi_1, a_1, \psi_1), \dots, (\varphi_k, a_k, \psi_k)\}$.

AAUL is evaluated on standard multi-agent Kripke models.

Definition 2. A model \mathcal{M} is a triple $\mathcal{M} = (W, R, V)$ where W is a set of states, $R : \mathcal{A} \rightarrow 2^{W \times W}$ assigns to each agent an accessibility relation and $V : \mathcal{P} \rightarrow 2^W$ is a valuation.

Note that we are using the class of all Kripke models. This is unlike APAL and GAL, which are typically considered on the class of S5 models.

Now, let us consider the semantics of AAUL. We start by giving the formal definition, after the definition we briefly discuss the intuition behind some of the operators.

Definition 3. Let $\mathcal{M} = (W, R, V)$ be a model and let $w \in W$. The satisfaction relation \models is given by

$$\begin{aligned}\mathcal{M}, w \models p & \quad \text{iff} \quad w \in V(p) \\ \mathcal{M}, w \models \neg\varphi & \quad \text{iff} \quad \mathcal{M}, w \not\models \varphi \\ \mathcal{M}, w \models (\varphi \wedge \psi) & \quad \text{iff} \quad \mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, w \models \psi \\ \mathcal{M}, w \models \Box_a\varphi & \quad \text{iff} \quad \mathcal{M}, v \models \varphi \text{ for each } v \text{ such that } (w, v) \in R(a) \\ \mathcal{M}, w \models [U]\varphi & \quad \text{iff} \quad (\mathcal{M} * U), w \models \varphi \\ \mathcal{M}, w \models [\uparrow]\varphi & \quad \text{iff} \quad \mathcal{M}, w \models [U]\varphi \text{ for each } U \in L_{AUL}\end{aligned}$$

where $(\mathcal{M} * U)$ is given by:

$$\begin{aligned}\mathcal{M} * U &= (W, R^U, V) \\ R^U(a) &= \{(v, v') \in R(a) \mid \exists (\varphi, a, \varphi') \in U : \\ & \quad (\mathcal{M}, v \models \varphi \text{ and } \mathcal{M}, v' \models \varphi')\}\end{aligned}$$

A full discussion of the applications of AAUL and of the intuitions behind the semantics of arrow updates and arbitrary arrow updates is outside the scope of this paper. For such a discussion, see [10] and [7]. However, in order to understand the undecidability proof it is important to understand the semantics

of AAUL. We therefore do provide a very brief explanation of the intuition behind and the semantics of AAUL.

Although our goal is to understand AAUL, it is useful to start by considering public announcements. We assume that the reader is familiar with public announcement logic, if not see for example [4]. A public announcement $[\psi]$ informs all agents that ψ is true. As a result, every possible world that the agents previously considered possible that does not satisfy ψ is rejected after the announcement, since it is incompatible with the new information. Semantically, this corresponds to a model \mathcal{M} being transformed into a model $\mathcal{M} * \psi$ where all $\neg\psi$ states of \mathcal{M} have been removed.

Like public announcements, arrow updates provide agents with new information. Unlike with public announcements, however, the new information provided by an arrow update can (i) differ per agent and (ii) depend on the current state. A typical example is a card game, where cards have been dealt face down. Now, agent a picks up her hand of cards and looks at it. Obviously, the information that a gains from this action is different than the information the other agents gain: a learns what her cards are whereas the other agents only learn that a now knows what her cards are. It is perhaps less obvious that the information that a gains also differs per state. Suppose that a has been dealt the 7 of Hearts. Then by looking at her cards a learns that she has the 7 of Hearts. If, on the other hand, a has been dealt the 8 of Clubs, then she learns that she has the 8 of Clubs. Learning that you have the 7 of Hearts is different from learning that you have the 8 of Clubs, so the information given to a depends on the state of the world.

With arrow updates we formalize the information that the agents gain in such a situation. In principle, we could do this in two ways: we could specify the things that are *incompatible* with the new information, or the things that are *compatible*. We choose to follow public announcements in this aspect, so just like $[\psi]$ says that the new information is compatible with ψ , we use an arrow update U to specify the information that is compatible with U . Since the information gained in an arrow update can depend on the agent and on the current state, we need triple (φ, a, ψ) . We call such triple *clauses*; they can be read as “if the current state satisfies φ , then the information provided to agent a is compatible with ψ .”

An arrow update U is a finite set clauses, $U = \{(\varphi_1, a_1, \psi_1), \dots, (\varphi_k, a_k, \psi_k)\}$ (where it is possible that $\varphi_i = \varphi_j$, $a_i = a_j$ or $\psi_i = \psi_j$ for $i \neq j$). This still leaves the decision of what to do if a state matches multiple clauses. Suppose, for example, that $(\varphi_1, a, \psi_1), (\varphi_2, a, \psi_2) \in U$ and that a state satisfies both φ_1 and φ_2 . There are several options for how to interpret this situation, we choose to interpret it disjunctively: if a state satisfies φ_1 and φ_2 , then any state that satisfies ψ_1 or ψ_2 is consistent with the new information.

On the semantical level, this means that $\mathcal{M} * U$ should contain exactly those arrows of \mathcal{M} that match at least one clause of U , where we say that $(w_1, w_2) \in R(a)$ matches (φ_1, a_1, ψ_1) if and only if $\mathcal{M}, w_1 \models \varphi_1$, $a = a_1$ and $\mathcal{M}, w_2 \models \psi_1$.

Arbitrary arrow updates then quantify over such arrow updates. However, in

order to avoid circularity we restrict this quantification to those arrow updates that do not themselves contain an arbitrary arrow update $[\Downarrow]$. So $\mathcal{M}, w \models [\Downarrow]\varphi$ if and only if $\mathcal{M}, w \models [U]\varphi$ for all $\varphi \in \mathcal{L}_{AUL}$.

3 Reducing the Tiling Problem

3.1 The Tiling Problem

We will prove the undecidability of AAUL by a reduction of the tiling problem. The tiling problem was introduced in [13] and can be defined as follows.

Definition 4. Let C be a finite set of colors. A *tile type* is a function $i : \{\text{north}, \text{south}, \text{east}, \text{west}\} \rightarrow C$.

An instance of the tiling problem is a finite set *types* of tile types. A solution to an instance of the tiling problem is a function $\text{tiling} : \mathbb{Z} \times \mathbb{Z} \rightarrow \text{types}$ such that, for every $(z_1, z_2) \in \mathbb{Z} \times \mathbb{Z}$,

$$\begin{aligned} \text{tiling}(z_1, z_2)(\text{north}) &= \text{tiling}(z_1, z_2 + 1)(\text{south}) \\ \text{tiling}(z_1, z_2)(\text{east}) &= \text{tiling}(z_1 + 1, z_2)(\text{west}). \end{aligned}$$

The tiling problem was shown to be undecidable in [5]. In fact, the tiling problem is co-RE complete. Therefore, by reducing the tiling problem to the satisfiability problem of AAUL, we show that the latter problem is co-RE hard. Whether AAUL's satisfiability problem is co-RE is not currently known.

3.2 Encoding the Tiling Problem in AAUL

We want to encode the tiling problem in AAUL. So for every instance *types* of the tiling problem we define a formula ψ_{types} of AAUL that is satisfiable if and only if *types* can tile the plane. The strategy for doing this is as follows.

We represent each point of $\mathbb{Z} \times \mathbb{Z}$ by a state. For every $i \in \text{types}$ we then use a propositional variable p_i to represent “the current state contains a tile of type i .” For every $c \in C$ we use propositional variables north_c (resp. $\text{south}_c, \text{east}_c, \text{west}_c$) to represent the northern (resp. southern, eastern, western) edge of the current tile having color c . Finally, we use relations *up*, *down*, *left* and *right* to represent one tile being above, below, to the left and to the right, respectively, of the current tile.

In addition to the states (n, m) that correspond to points in $\mathbb{Z} \times \mathbb{Z}$, we also use an auxiliary state s_0 . This state s_0 is not part of the grid, and does not contain any tile. Instead, it is the state where ψ_{types} will be evaluated. We therefore also refer to s_0 as the origin state. In order to distinguish s_0 from the states that are part of the grid we use the propositional variable p , which holds on s_0 but not on any (n, m) .

Now, given any state (n, m) , it is relatively easy to check whether the constraints of a tiling are satisfied locally. For example, $\bigvee_{i \in \text{types}} p_i \wedge \bigwedge_{i \neq j \in \text{types}} \neg(p_i \wedge p_j)$ holds if and only if the current state has exactly one type of tile, and

$\bigwedge_{c \in C} (north_c \rightarrow \Box_{up} south_c)$ holds if and only if the northern color of the current tile matches the southern color of the tile above.

Making sure that the *global* constraints of a tiling are satisfied is harder, though. We do this in the following way. Firstly, we take a relation b , and force it to act as a kind of transitive closure¹ over *up*, *down*, *left* and *right*. So while $\bigvee_{i \in types} p_i \wedge \bigwedge_{i \neq j \in types} \neg(p_i \wedge p_j)$ says that the current state has exactly one tile type, the formula $\Box_b \bigvee_{i \in types} p_i \wedge \bigwedge_{i \neq j \in types} \neg(p_i \wedge p_j)$ says that *all* states (except the current one²) have exactly one tile type. Secondly, we enforce a grid-like structure onto the domain.

With the above in mind, let us define the formula ψ_{types} .

Definition 5. Let $types$ be an instance of the tiling problem. The formula ψ_{types} is given by

$$\begin{aligned} \psi_{types} := & \psi_1 \wedge \psi_2 \wedge \bigwedge_{x \in D} (\psi_{3,x} \wedge \psi_{4,x} \wedge propd_x \wedge return_x) \\ & \wedge inverse \wedge commute \\ & \wedge one_tile \wedge one_color \wedge tile_colors \wedge tile_match \end{aligned}$$

where

$$\begin{aligned} D := & \{up, down, left, right\} \\ \psi_1 := & ref_a \wedge p \wedge \Diamond_b \top \wedge \Box_b \neg p \\ \psi_2 := & \Box_b (ref_a \wedge \Diamond_b p) \wedge [\uparrow](\Diamond_a \top \rightarrow \Box_b \Box_b \Diamond_a \top) \\ \psi_{3,x} := & \Box_b (\Diamond_x (\neg p \wedge ref_a \wedge \Diamond_b p) \wedge [\uparrow](\Diamond_x \Diamond_a \top \rightarrow \Box_x \Diamond_a \top)) \\ \psi_{4,x} := & [\uparrow](\Diamond_a \top \rightarrow \Box_b \Box_x \Box_b \Diamond_a \top) \\ ref_a := & \Diamond_a \Diamond_a \top \wedge [\uparrow] \neg \Diamond_a \Box_a \perp \\ propd_x := & \Box_b [\uparrow] ((\Box_a \perp \wedge \Diamond_x \Diamond_a \top \wedge \Diamond_b (\Diamond_b \top \wedge \Box_b \Diamond_a \top) \wedge \\ & \langle \uparrow \rangle (\Diamond_x \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \perp)) \rightarrow [U_x] \langle \uparrow \rangle (\Diamond_x \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \perp)) \\ U_x := & (p \vee \Box_a \perp, b, \top), (\top, a, \top), (\Box_a \perp, x, \top) \\ return_x := & \Box_b \langle \uparrow \rangle (\Box_a \perp \wedge \Diamond_b \top \wedge (\Diamond_a \top \wedge \Diamond_b (\Diamond_b \top \wedge \Box_b \Diamond_a \top) \wedge \\ & [\uparrow](\Diamond_a \top \rightarrow \Box_b \Box_b \Diamond_a \top))) \\ inverse := & \Box_b [\uparrow] (\Box_a \perp \rightarrow (\Box_{up} \Box_{down} \Box_a \perp \wedge \Box_{down} \Box_{up} \Box_a \perp \\ & \wedge \Box_{left} \Box_{right} \Box_a \perp \wedge \Box_{right} \Box_{left} \Box_a \perp)) \\ commute := & \Box_b [\uparrow] \bigwedge_{(x,y) \in E} (\Diamond_x \Diamond_y \Box_a \perp \rightarrow \Box_y \Box_x \Box_a \perp) \\ E := & \{(up, left), (up, right), (down, left), (down, right), \\ & (left, up), (left, down), (right, up), (right, down)\} \end{aligned}$$

¹The precise properties of b are more complicated than this, but for the purpose of this informal introduction to the proof it suffices to think of it as a transitive closure.

²Recall that ψ_{types} will be evaluated in s_0 , so the “current state” in question is s_0 , which does not have a tile type.

$$\begin{aligned}
one_tile &:= \Box_b \left(\bigvee_{i \in tiles} p_i \wedge \bigwedge_{i \neq j \in tiles} \neg(p_i \wedge p_j) \right) \\
one_color &:= \Box_b \bigwedge_{c \in C} (north_c \rightarrow \bigwedge_{d \in C \setminus \{c\}} \neg north_d) \wedge \\
&\quad \Box_b \bigwedge_{c \in C} (south_c \rightarrow \bigwedge_{d \in C \setminus \{c\}} \neg south_d) \wedge \\
&\quad \Box_b \bigwedge_{c \in C} (east_c \rightarrow \bigwedge_{d \in C \setminus \{c\}} \neg east_d) \wedge \\
&\quad \Box_b \bigwedge_{c \in C} (west_c \rightarrow \bigwedge_{d \in C \setminus \{c\}} \neg west_d) \\
tile_colors &:= \Box_b \bigwedge_{i \in tiles} (p_i \rightarrow (north_{i(north)} \\
&\quad \wedge south_{i(south)} \wedge east_{i(east)} \wedge west_{i(west)})) \\
tile_match &:= \Box_b \bigwedge_{c \in C} ((north_c \rightarrow \Box_{up} south_c) \wedge (west_c \rightarrow \Box_{left} east_c))
\end{aligned}$$

The first line of ψ_{types} guarantees that each state other than the origin has exactly one successor for each direction $x \in D$, and that b is transitive over all $x \in D$ (by which we mean: if $(s_0, s_1) \in R(b)$ and $(s_1, s_2) \in R(x)$, then $(s_0, s_2) \in R(b)$). The second line of ψ_{types} then guarantees that the four directions form a grid. Finally, the third line of ψ_{types} guarantees that the grid is tiled in an appropriate way.

We need to show that ψ_{types} is satisfiable if and only if $types$ can tile $\mathbb{Z} \times \mathbb{Z}$. We start by showing that if such a tiling exists, then ψ_{types} is satisfiable.

Lemma 1. *Suppose $types$ can tile $\mathbb{Z} \times \mathbb{Z}$. Then ψ_{types} is satisfiable.*

Proof. Let $tiling$ be the tiling, let $p_{n,m} \in \mathcal{P}$ for every $n, m \in \mathbb{Z}$ and let $\mathcal{M} = (S, R, V)$ be the following, quite straightforward, encoding of $tiling$:

- $S = (\mathbb{Z} \times \mathbb{Z}) \cup s_0$
- $R(a) = \{(s, s) \mid s \in S\}$
- $R(b) = \{(s_0, (n, m)) \mid n, m \in \mathbb{Z}\} \cup \{((n, m), s_0) \mid n, m \in \mathbb{Z}\}$
- $R(up) = \{((n, m), (n, m + 1)) \mid n, m \in \mathbb{Z}\}$
- $R(down) = \{((n, m), (n, m - 1)) \mid n, m \in \mathbb{Z}\}$
- $R(left) = \{((n, m), (n - 1, m)) \mid n, m \in \mathbb{Z}\}$
- $R(right) = \{((n, m), (n + 1, m)) \mid n, m \in \mathbb{Z}\}$
- $V(p) = \{s_0\}$
- $V(p_i) = \{(n, m) \mid tiling(n, m) = i\}$ for $i \in tiles$

- $V(north_c) = \{(n, m) \mid ((tiling)(n, m))(north) = c\}$ for $c \in C$
- $V(south_c) = \{(n, m) \mid ((tiling)(n, m))(south) = c\}$ for $c \in C$
- $V(east_c) = \{(n, m) \mid ((tiling)(n, m))(east) = c\}$ for $c \in C$
- $V(west_c) = \{(n, m) \mid ((tiling)(n, m))(west) = c\}$ for $c \in C$
- $V(p_{n,m}) = \{(n, m)\}$

As mentioned above, the state s_0 is special: it is the origin state, and the only state that does not have a tile type associated with it. The propositional variable p is used to identify this special state. We will show that $\mathcal{M}, s_0 \models \psi_{types}$.

Note that $R(a)$ is the identity relation. As a result, there can be no arrow update that retains the a -arrow from s to some state s' but removes the a -arrow from s' . So $\mathcal{M} \models \Diamond_a \Diamond_a \top \wedge [\uparrow] \neg \Diamond_a \Box_a \perp$ and therefore, by definition, $\mathcal{M} \models ref_a$.

Now, consider the pointed model \mathcal{M}, s_0 . It is straightforward to verify that $\mathcal{M}, s_0 \models p \wedge \Diamond_b \top \wedge \Box_b \neg p$. Because we already determined that $\mathcal{M}, s_0 \models ref_a$, this suffices to show that $\mathcal{M}, s_0 \models \psi_1$.

Additionally, $\mathcal{M} \models ref_a$ together with the fact that there are b -arrows from s_0 to every (n, m) and from every (n, m) to s_0 , which satisfies p , implies that $\mathcal{M}, s_0 \models \Box_b (ref_a \wedge \Diamond_b p)$. Furthermore, every state (n, m) has only one outgoing b -arrow, namely the one to s_0 . So for any arrow update U , if U retains the a -arrow on s_0 then it must also retain the a -arrow on any state that is b -reachable from a state (n, m) , since it is the same a -arrow. So $\mathcal{M}, s_0 \models [\uparrow](\Diamond_a \top \rightarrow \Box_b \Box_b \Diamond_a \top)$. We have shown that \mathcal{M}, s_0 satisfies both conjuncts of ψ_2 , so $\mathcal{M}, s_0 \models \psi_2$.

Now, let us look at $\psi_{3,x}$. For every direction x and every (n, m) , there is an x -arrow from (n, m) to some (n', m') . Like (n, m) , this (n', m') satisfies $\neg p \wedge ref_a \wedge \Diamond_b p$, so we have $\mathcal{M}, (n, m) \models \Diamond_x (\neg p \wedge ref_a \wedge \Diamond_b p)$. For every direction there is only one such successor (n', m') . This implies that every arrow update either removes the a -arrow from all x -successors of (n, m) , or it retains the a -arrow of all such successors. This implies that $\mathcal{M}, (n, m) \models [\uparrow](\Diamond_x \Diamond_a \top \rightarrow \Box_x \Diamond_a \top)$. This is true for any (n, m) , so we have $\mathcal{M}, s_0 \models \psi_{3,x}$ for every direction $x \in D$.

All states (n, m) have a b -arrow to the same state s_0 . Once again we use the fact that every arrow update either eliminates or retains the a -arrow on this single world, it cannot do both. It follows that $\mathcal{M}, s_0 \models [\uparrow](\Diamond_a \top \rightarrow \Box_b \Box_x \Box_b \Diamond_a \top)$, so, by definition, $\mathcal{M}, s_0 \models \psi_{4,x}$.

Now we get to the more difficult part, where we need to show that s_0 satisfies the rather complicated formula $propd_x$, for every direction x . For ease of notation, we will show that $propd_{right}$ holds, the proof for the other directions is similar. Fix any (n, m) , and let U be any update such that the antecedent of the implication in $propd_{right}$ holds after U , i.e.

$$\begin{aligned} \mathcal{M} * U, (n, m) \models & \Box_a \perp \wedge \Diamond_{right} \Diamond_a \top \wedge \Diamond_b (\Diamond_b \top \wedge \Box_b \Diamond_a \top) \wedge \\ & (\uparrow)(\Diamond_{right} \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \perp) \end{aligned}$$

The assumption that this antecedent holds places some restrictions on U . The first conjunct says that the a -arrow on (n, m) is removed. The second conjunct says that the *right*-arrow from (n, m) to $(n + 1, m)$ is retained, and so is the a -arrow on $(n + 1, m)$. The third conjunct says that the b -arrow from (n, m) to s_0 is retained, that at least one b -arrow from s_0 to some (n', m') is retained and that every (u, v) that is b -accessible from s_0 still has its a -arrow. The final conjunct then says that there is some arrow update, call it U' , that

- retains the *right*-arrow from (n, m) to $(n + 1, m)$ and the a -arrow on $(n + 1, m)$,
- retains the b -arrow from (n, m) to s_0 and the b -arrow from s_0 to some state (u, v) and
- removes the a -arrow on (u, v) .

In particular, the fact that the a -arrow on (u, v) is removed while that on $(n + 1, m)$ is retained shows that, $(u, v) \neq (n + 1, m)$.

For any such U , we need to show that the consequent is true in (n, m) , i.e. that

$$\mathcal{M} * U, (n, m) \models [U_{\text{right}}](\Downarrow)(\Diamond_{\text{right}}\Diamond_a\top \wedge \Diamond_b\Diamond_b\Box_a\perp).$$

Recall that $U_{\text{right}} = (p \vee \Box_a\perp, b, \top), (\top, a, \top), (\Box_a\perp, \text{right}, \top)$. So U_{right} retains all b -arrows from s_0 (since $\mathcal{M} * U, s_0 \models p$), the b -arrow from (n, m) to s_0 and the *right*-arrow from (n, m) to $(n + 1, m)$ (since $\mathcal{M} * U, (n, m) \models \Box_a\perp$) and every a -arrow—and therefore in particular the a -arrows on (n, m) and (u, v) . Because $(u, v) \neq (n + 1, m)$, the atom $p_{(n+1, m)}$ holds in $(n + 1, m)$ but not in (u, v) . The update $U'' := (\top, \text{right}, \top), (\top, b, \top), (p_{(n+1, m)}, a, \top)$ therefore retains all *right* and b arrows as well as the a -arrow on $(n + 1, m)$ while removing the a -arrow on (u, v) . Since $\mathcal{M} * U$, and therefore also $(\mathcal{M} * U) * U''$, contains a b -arrow from (n, m) to s_0 and from s_0 to (u, v) as well as a *right*-arrow from (n, m) to $(n + 1, m)$ we have $(\mathcal{M} * U) * U_{\text{right}}, (n, m) \models [U''](\Diamond_{\text{right}}\Diamond_a\top \wedge \Diamond_b\Diamond_b\Box_a\perp)$ and therefore $\mathcal{M} * U, (n, m) \models [U_{\text{right}}](\Downarrow)(\Diamond_{\text{right}}\Diamond_a\top \wedge \Diamond_b\Diamond_b\Box_a\perp)$, which was to be shown.

This completes the proof that $\mathcal{M}, s_0 \models \text{propd}_{\text{right}}$. We continue by showing that $\mathcal{M}, s_0 \models \text{return}_x$. Again, we show that $\mathcal{M}, s_0 \models \text{return}_{\text{right}}$, the other directions can be proven similarly. The formula $\text{return}_{\text{right}}$ holds in \mathcal{M}, s_0 if

$$\begin{aligned} \mathcal{M}, (n, m) \models & (\Downarrow)(\Box_a\perp \wedge \Diamond_b\top \wedge \Diamond_{\text{right}}(\Downarrow)(\Diamond_a\top \wedge \Diamond_b(\Diamond_b\top \wedge \Box_b\Diamond_a\top) \wedge \\ & (\Downarrow)(\Diamond_a\top \rightarrow \Box_b\Box_b\Diamond_a\top))) \end{aligned}$$

for every (n, m) . Let $U_1 := (\top, \text{right}, \top), (\top, b, p \vee p_{n+1, m}), (p_{(n+1, m)}, a, \top)$. The only b -arrows that are retained by U_1 are those that go to s_0 or to $(n + 1, m)$. Because the only remaining b -arrow from s_0 is to $(n + 1, m)$, every arrow update that retains the a -arrow on $(n + 1, m)$ must do so in every b -successor of every b -successor of $(n + 1, m)$, since $(n + 1, m)$ is the only such b - b -successor. So

$\mathcal{M} * U_1, (n+1, m) \models [\Downarrow](\Diamond_a \top \rightarrow \Box_b \Box_b \Diamond_a \top)$. Furthermore, U_1 retains the a -arrow on $(n+1, m)$ so, again using the fact that $(n+1, m)$ is the only remaining b -successor of s_0 , we also have $\mathcal{M} * U_1, (n+1, m) \models \Diamond_a \top \wedge \Diamond_b(\Diamond_b \top \wedge \Box_b \Diamond_a \top)$.

Putting these conjuncts together, and using the fact that $\models \chi \rightarrow \langle \Downarrow \rangle \chi$ for every χ , we get

$$\mathcal{M} * U_1, (n+1, m) \models \langle \Downarrow \rangle (\Diamond_a \top \wedge \Diamond_b(\Diamond_b \top \wedge \Box_b \Diamond_a \top) \wedge [\Downarrow](\Diamond_a \top \rightarrow \Box_b \Box_b \Diamond_a \top)).$$

The update U_1 also eliminates the a -arrow on (n, m) while retaining the b -arrow from (n, m) to s_0 and the *right*-arrow from (n, m) to $(n+1, m)$, so

$$\begin{aligned} \mathcal{M} * U_1, (n, m) \models & \Box_a \perp \wedge \Diamond_b \top \wedge \Diamond_{right} \langle \Downarrow \rangle (\Diamond_a \top \wedge \Diamond_b(\Diamond_b \top \wedge \Box_b \Diamond_a \top) \wedge \\ & [\Downarrow](\Diamond_a \top \rightarrow \Box_b \Box_b \Diamond_a \top)). \end{aligned}$$

We have $U_1 \in \mathcal{L}_{AUL}$, so

$$\begin{aligned} \mathcal{M}, (n, m) \models & \langle \Downarrow \rangle (\Box_a \perp \wedge \Diamond_b \top \wedge \Diamond_{right} \langle \Downarrow \rangle (\Diamond_a \top \wedge \Diamond_b(\Diamond_b \top \wedge \Box_b \Diamond_a \top) \wedge \\ & [\Downarrow](\Diamond_a \top \rightarrow \Box_b \Box_b \Diamond_a \top))), \end{aligned}$$

which was to be shown in order to prove that $\mathcal{M}, s_0 \models return_{right}$.

We continue with *inverse*. In \mathcal{M} , the relations *up* and *down* are each others inverses, as are *left* and *right*. Furthermore, all four direction relations are functions. It follows immediately that, for every (n, m) , we have $\mathcal{M}, (n, m) \models [\Downarrow](\Box_a \perp \rightarrow \Box_{right} \Box_{left} \Box_a \perp)$, and similarly for the other combinations of directions. So we have $\mathcal{M}, s_0 \models inverse$.

Similarly, in \mathcal{M} we have $R(right) \circ R(up) = R(up) \circ R(right)$, and the same for the other directions. It follows that $\mathcal{M}, s_0 \models \Box_b [\Downarrow] \bigwedge_{(x,y) \in E} (\Diamond_x \Diamond_y \Box_a \perp \rightarrow \Box_y \Box_x \Box_a \perp)$.

The last four conjuncts of ψ_{types} simply encode the fact that *tiling* is a tiling on $\mathbb{Z} \times \mathbb{Z}$, so \mathcal{M}, s_0 satisfies those as well. \square

Left to show is that if ψ_{types} is satisfiable, then *types* can tile the plane. We do this by showing that any model where ψ_{types} is satisfied looks like the model \mathcal{M} that we constructed above. There are some differences, certainly. For example, we cannot enforce that a state (n, m) has exactly one x -successor for every direction x , only that it has at least one x -successor and that all its x -successors are indistinguishable. Such differences are not relevant to the existence of a tiling, however.

Lemma 2. *If ψ_{types} is satisfiable, then there is a tiling on $\mathbb{Z} \times \mathbb{Z}$ for types.*

Proof. Suppose ψ_{types} is satisfiable. Then there is some pointed model \mathcal{M}, s_0 such that $\mathcal{M}, s_0 \models \psi_{types}$.

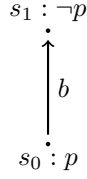
First, consider any state s such that \mathcal{M}, s satisfies ref_a . Then there is some s' that is a -accessible from s . Furthermore, by $[\Downarrow] \neg \Diamond_a \Box_a \perp$, there is no arrow update that retains the a -arrow from s to s' while removing any a -arrows from s' . It follows that there is no modal formula φ that holds on s but not on s' ,

since otherwise (φ, a, \top) would have been such an arrow update. So any state s that satisfies ref_a has an a -arrow to a state that is indistinguishable from s , and it only has a -arrows to such indistinguishable states.

In order to make the proof easier to read, it is convenient to draw figures of \mathcal{M} as we use ψ_{types} to construct it. In these figures, we use the following rules for simplification:

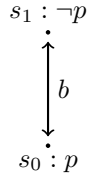
1. We do not draw the entire model, but only those parts of it that are of interest for the part of the proof illustrated by the figure. So unless explicitly stated otherwise, the lack of a drawn arrow between two states does not always indicate that there is no arrow between them.
2. When two states are modally indistinguishable, we draw them as if they were one state. So, strictly speaking, when we draw a state s we mean an equivalence class $[s]$ of states.
3. Most states that we consider satisfy ref_a . Together with the second rule of simplification, this means we could draw a reflexive a -arrow on those worlds. For simplicity of presentation we do not draw these arrows.

Now, suppose $\mathcal{M}, s_0 \models \psi_{types}$. The first conjunct of ψ_{types} states that $\mathcal{M}, s_0 \models ref_a \wedge p \wedge \Diamond_b \top \wedge \Box_b \neg p$. This means that, so far, our model looks something like the following.



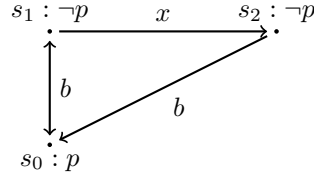
Next, consider ψ_2 , the second conjunct of ψ_{types} , which states that $\mathcal{M}, s_0 \models \Box_b(ref_a \wedge \Diamond_b p) \wedge [\uparrow](\Diamond_a \top \rightarrow \Box_b \Box_b \Diamond_a \top)$. In particular, it implies that any state s_1 that is b -accessible from s_0 satisfies ref_a , and has a b -accessible p state.

Suppose towards a contradiction that there is any b -successor s' of s_1 that is modally distinguishable from s_0 . Then there is some φ that holds on s_0 but not on s' . Consider then the update $U = (\varphi, a, \top), (\top, b, \top)$. We have $\mathcal{M}, s_0 \models [U](\Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \perp)$, contradicting $\mathcal{M}, s_0 \models [\uparrow](\Diamond_a \top \rightarrow \Box_b \Box_b \Diamond_a \top)$. Our assumption that such distinguishable s' exists was therefore false, so our model looks like this:



Now, let us consider $\psi_{3,x}$, which tells us that $\mathcal{M}, s_0 \models \Box_b(\Diamond_x(\neg p \wedge \text{ref}_a \wedge \Diamond_b p) \wedge [\uparrow](\Diamond_x \Diamond_a \top \rightarrow \Box_x \Diamond_a \top))$ for every direction $x \in D$. Firstly, from $\mathcal{M}, s_1 \models \Diamond_x(\neg p \wedge \text{ref}_a \wedge \Diamond_b p)$ we learn that there is a state s_2 that is accessible from s_1 , and that this s_2 satisfies $\neg p$, ref_a and $\Diamond_b p$. Furthermore, by $\mathcal{M}, s_0 \models [\uparrow](\Diamond_x \Diamond_a \top \rightarrow \Box_x \Diamond_a \top)$ we know that this s_2 is unique (up to modal indistinguishability), since otherwise there would be some arrow update that retains the a -arrow on s_2 but removes it from some other x -successor of s_1 .

Let us then look at the conjunct $\psi_{4,x}$, which states that $\mathcal{M}, s_0 \models [\uparrow](\Diamond_a \top \rightarrow \Box_b \Box_x \Box_b \Diamond_a \top)$. This tells us that every b -successor of s_2 must be indistinguishable from s_0 , as otherwise there would be an arrow update removing the a -arrow from such a successor while retaining the a -arrow on s_0 . Our drawing of the model therefore becomes as follows.



Now, we get to the hard part. The formula propd_x , which is the fifth conjunct of ψ_{types} , tells us that

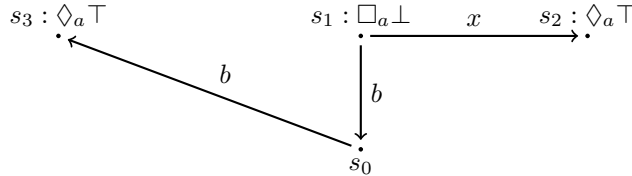
$$\mathcal{M}, s_0 \models \Box_b[\uparrow]((\Box_a \perp \wedge \Diamond_x \Diamond_a \top \wedge \Diamond_b(\Diamond_b \top \wedge \Box_b \Diamond_a \top) \wedge [\uparrow](\Diamond_x \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \perp)) \rightarrow [U_x](\uparrow)(\Diamond_x \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \perp))$$

where

$$U_x := (p \vee \Box_a \perp, b, \top), (\top, a, \top), (\Box_a \perp, x, \top).$$

The formula starts with \Box_b , which takes us to any b -successor of s_0 . Let us assume without loss of generality that this is the s_1 we drew earlier. In s_1 , it must be the case that any update U_1 that makes $(\Box_a \perp \wedge \Diamond_x \Diamond_a \top \wedge \Diamond_b(\Diamond_b \top \wedge \Box_b \Diamond_a \top) \wedge [\uparrow](\Diamond_x \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \perp))$ true also makes $[U_x](\uparrow)(\Diamond_x \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \perp)$ true. Let us look at what $\mathcal{M} * U_1$ would look like. Firstly, we have $\mathcal{M} * U_1, s_1 \models \Box_a \perp \wedge \Diamond_x \Diamond_a \top$. So the a -arrow on s_1 is removed, while the x -arrow to s_2 and the a -arrow on s_2 are retained.

Furthermore, $\mathcal{M} * U_1, s_1 \models \Diamond_b(\Diamond_b \top \wedge \Box_b \Diamond_a \top)$, so the b -arrow from s_1 to s_0 is retained, at least one b -arrow from s_0 is retained and every b -arrow from s_0 that still exists points to a \Diamond_a state. So far, the situation in $\mathcal{M} * U_1$ can be drawn as in the following diagram, where s_3 is some state b -accessible from s_0 .



Finally, $\mathcal{M} * U_1, s_1 \models \langle \downarrow \rangle (\Diamond_x \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \perp)$. So there is some arrow update U_2 that retains the a -arrow on s_2 while removing the a -arrow on s_3 . Such U_2 exists if and only if s_2 and s_3 are modally distinguishable in $\mathcal{M} * U_1$. The formula propd_x states that for every such U_1 , we must have $\mathcal{M} * U_1, s_1 \models [U_x] \langle \downarrow \rangle (\Diamond_x \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \perp)$. The update U_x is designed in such a way that it removes all arrows, other than the reflexive a -arrow, from s_2 and s_3 while retaining the x arrow from s_1 to s_2 as well as the b -arrows from s_1 to s_0 and from s_0 to s_3 . So, using the simplification rules, the drawing of $(\mathcal{M} * U_1) * U_x$ is actually the same as the drawing of $\mathcal{M} * U_1$ given above.

However, there is a difference. In $\mathcal{M} * U_1$, we did not draw any arrows from s_2 and s_3 because there are not guaranteed to be such arrows. In $(\mathcal{M} * U_1) * U_x$ we did not draw any such arrows because they are guaranteed not to exist.

We have $(\mathcal{M} * U_1) * U_x, s_1 \models \langle \downarrow \rangle \Diamond_x \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \perp$, which is the case if and only if s_2 and s_3 are distinguishable in $(\mathcal{M} * U_1) * U_x$. Since, in $(\mathcal{M} * U_1) * U_x$, s_2 and s_3 have no outgoing arrows other than the a -arrow to a state that is indistinguishable from them, it follows that s_2 and s_3 must be propositionally distinguishable.

In summary, propd_x states that, for any arrow update U_1 , if $\mathcal{M} * U_1$ matches the drawing shown above and s_2 is distinguishable (in $\mathcal{M} * U_1$) from some b -successor (in $\mathcal{M} * U_1$) s_3 of s_0 , then it is propositionally distinguishable from such a successor.

Furthermore, we can show that the states also have to be propositionally distinguishable if they are distinguishable in \mathcal{M} (as opposed to $\mathcal{M} * U_1$). Suppose that U_1 is such that $\mathcal{M} * U_1$ matches the drawing and that s_2 is distinguishable from s_3 in \mathcal{M} . Now, let χ be any modal formula that distinguishes between s_2 and s_3 and (by negating if necessary) assume that χ holds on s_1 . We distinguish between three cases:

- Suppose U_1 removes the b -arrows from both s_2 and s_3 . Then let $U'_1 := U_1 \cup \{(\chi \wedge \neg p, a, \top)\}$.
- Suppose U_1 removes the b -arrow from one of s_2 and s_3 , while retaining the other. Then let $U'_1 := U_1$.
- Suppose U_1 retains the b -arrows on both s_2 and s_3 . Then let U'_1 be the update obtained by replacing every clause $(\varphi, b, \psi) \in U_1$ by $(\varphi \wedge (p \vee \chi), b, \psi)$.

In any of the three cases, $\mathcal{M} * U'_1$ matches the figure and, furthermore, exactly one of s_2 and s_3 has a b -successor in $\mathcal{M} * U'_1$. In particular, s_2 and s_3 are distinguishable in $\mathcal{M} * U'_1$. It follows that s_2 and s_3 are propositionally distinguishable.

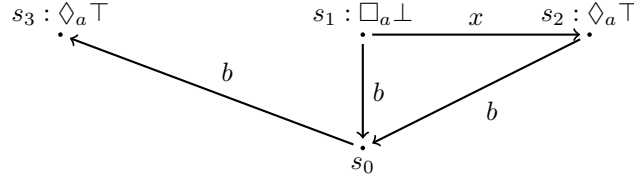
Summarizing once again, using this new observation: for any arrow update U_1 , if $\mathcal{M} * U_1$ matches the drawing and s_2 is distinguishable (in \mathcal{M}) from some b -successor (in $\mathcal{M} * U_1$) s_3 of s_0 , then s_2 is propositionally distinguishable from such a b -successor.

Next, we look at $return_x$. We have

$$\mathcal{M}, s_0 \models \Box_b(\uparrow)(\Box_a\perp \wedge \Diamond_b\top \wedge \Diamond_x(\Diamond_a\top \wedge \Diamond_b(\Diamond_b\top \wedge \Box_b\Diamond_a\top) \wedge [\uparrow](\Diamond_a\top \rightarrow \Box_b\Box_b\Diamond_a\top))).$$

Again, this starts with a \Box_b operator, so we go to any b -successor of s_0 and assume without loss of generality that this successor is s_1 . Then there is some update U_1 such that $\mathcal{M} * U_1, s_1 \models (\Box_a\perp \wedge \Diamond_b\top \wedge \Diamond_x(\Diamond_a\top \wedge \Diamond_b(\Diamond_b\top \wedge \Box_b\Diamond_a\top) \wedge [\uparrow](\Diamond_a\top \rightarrow \Box_b\Box_b\Diamond_a\top)))$.

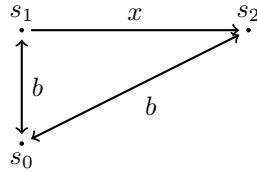
We will discuss the $[\uparrow](\Diamond_a\top \rightarrow \Box_b\Box_b\Diamond_a\top)$ part of this formula later. For now, note that the other parts of the formula state that the model $\mathcal{M} * U_1$ can be drawn as follows.



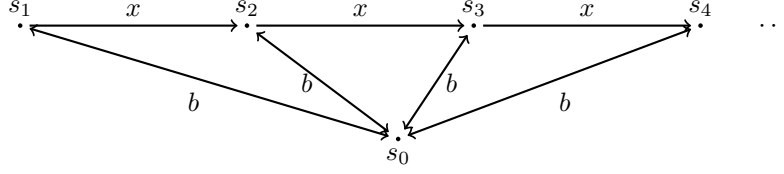
Now recall the diagram drawn further above, when discussing $propd_x$. The only difference between the two diagrams is that the newer one has an extra b -arrow from s_1 to s_0 . By the conventions with which we draw these diagrams, this means the newer diagram is an instance of the older one. So by the fact that $\mathcal{M}, s_0 \models propd_x$, we know that if s_2 and s_3 are distinguishable in \mathcal{M} , then they are propositionally distinguishable so, in particular, that they are distinguishable in $\mathcal{M} * U_1$.

Now, we return to the subformula $[\uparrow](\Diamond_a\top \rightarrow \Box_b\Box_b\Diamond_a\top)$, which has to hold in s_2 . It implies that we cannot retain the a -arrow on s_2 while removing the one on s_3 . So s_2 and s_3 are indistinguishable in $\mathcal{M} * U_1$. We already determined that this implies that s_2 and s_3 are indistinguishable in \mathcal{M} as well.

In particular, this implies that there is some state s_3 that is b -reachable from s_0 and that is indistinguishable from s_2 (in \mathcal{M}). So b is what could be called *indistinguishable-transitive* over x , i.e. for every $(s_0, s_1) \in R(b)$ and every $(s_1, s_2) \in R(x)$ there is some s_3 that is modally indistinguishable from s_2 such that $(s_0, s_3) \in R(b)$. By the convention to draw indistinguishable worlds as being the same world, this means our diagram of \mathcal{M} simplifies to the following:



Furthermore, this holds for any b -successor s_1 of s_0 . So there is also a unique (up to indistinguishability) x -successor s_3 of s_2 that is b -reachable from s_1 , a unique x -successor s_4 of s_3 that is b -reachable from s_0 and so on.



Also note that this holds for every direction $x \in D$. So we are approaching the grid-like structure that we need, with s_0 being a special point that lies outside the grid.

The next two formulas enforce the grid-like structure, by placing restrictions on certain combinations of directions. Firstly, consider *inverse*. We have

$$\mathcal{M}, s_0 \models \Box_b[\uparrow](\Box_a \perp \rightarrow (\Box_{up}\Box_{down}\Box_a \perp \wedge \Box_{down}\Box_{up}\Box_a \perp \wedge \Box_{left}\Box_{right}\Box_a \perp \wedge \Box_{right}\Box_{left}\Box_a \perp))$$

Simply put, this formula states that, for every b -successor s_1 of s_0 , the unique *down*-successor of the *up*-successor of s_1 is indistinguishable from s_1 , and similar for the other combinations of opposite directions. So in \mathcal{M} the opposite directions act as each other's inverses.

Now, consider *commute*. We have

$$\mathcal{M}, s_0 \models \Box_b[\uparrow] \bigwedge_{(x,y) \in E} (\Diamond_x \Diamond_y \Box_a \perp \rightarrow \Box_y \Box_x \Box_a \perp)$$

and $E = \{(up, left), (up, right), (down, left), (down, right), (left, up), (left, down), (right, up), (right, down)\}$. This formula simply states that the *up*-successor of the *left*-successor of s_1 is indistinguishable from the *left*-successor of its *up*-successor, and the same for the other combinations of directions.

Putting all of the above together, we learn that the b -successors of s_0 form a two-dimensional grid.³ The remaining conjuncts of ψ_{types} straightforwardly encode that this grid is tiled by the set *types*.

The formula *one_tile* states that every point in the grid satisfies exactly one propositional variable p_i with $i \in types$. The formula *one_color* states that every edge of every point in the grid has at most one color for each direction. The formula *tile_colors* guarantees that if a point satisfies p_i then the edge colors of that point are the edge colors of tile i . Finally, the formula *tile_match* guarantees that the colors of opposing edges of neighboring states match.

All in all, this shows that if ψ_{types} is satisfiable, then the set *types* can tile the plane. \square

Theorem 1. *The satisfiability problem of AAUL is co-RE hard.*

Proof. Given an instance *types* of the tiling problem, the formula ψ_{types} is computable. Furthermore, Lemmas 1 and 2 show that ψ_{types} is satisfiable if and only if *types* can tile the plane. The tiling problem is known to be co-RE complete [5], therefore the satisfiability problem of AAUL is co-RE hard. \square

³Or, to be slightly more precise: at least one two-dimensional grid. There may be multiple disconnected grids.

4 Conclusion

We have shown that the satisfiability of AAUL is uncomputable, like that of similar logics such as APAL [8] and GAL [2]. It is not currently known whether the satisfiability problem of AAUL is co-RE. Typically, one would show that a satisfiability problem is co-RE by providing an axiomatization for the logic, thereby showing the validities of the logic to be RE. However, while there are known axiomatizations for AAUL, APAL and GAL, these axiomatizations are infinitary and therefore cannot be used to enumerate the valid formulas of the logics in question.⁴

One interesting direction for future research is therefore to determine whether the satisfiability problems of AAUL, APAL and GAL are co-RE complete, and whether these logics admit finitary axiomatizations.

In principle, the proof that we gave for the undecidability of AAUL applies only to the satisfiability problem when considered over the class of all Kripke models. However, we believe that the proof can be adapted to work for the satisfiability problem with respect to other common classes of models such as KD45 and S5.

References

- [1] T. Ågotnes, P. Balbiani, H. van Ditmarsch, and P. Seban. Group announcement logic. *Journal of Applied Logic*, 8(1):62 – 81, 2010.
- [2] Thomas Ågotnes, Hans van Ditmarsch, and Tim French. The undecidability of group announcements. In Ana Bazzan Michael Huhns Alessio Lomuscio, Paul Scerri, editor, *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014)*, pages 893–900, 2014.
- [3] P. Balbiani, A. Baltag, H. van Ditmarsch, A. Herzig, T. Hoshi, and T. de Lima. ‘Knowable’ as ‘known after an announcement’. *Review of Symbolic Logic*, 1(3):205–334, 2008.
- [4] Alexandru Baltag, Lawrence Moss, and Sławomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In I. Gilboa, editor, *Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge*, pages 43–56. Morgan Kaufmann Publishers Inc., 1998.
- [5] Robert Berger. *The Undecidability of the Domino Problem*. Number 66 in Memoirs of the American Mathematical Society. 1966.

⁴Finitary axiomatizations for APAL and GAL were proposed, in [3] and [1] respectively, but these were later shown to be unsound.

- [6] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. Arbitrary arrow update logic. In *Advances in Modal Logic 2014, short presentations*, 2014.
- [7] Hans van Ditmarsch, Wiebe van der Hoek, Barteld Kooi, and Louwe B. Kuijer. Arbitrary arrow update logic, 2016. Submitted for publication.
- [8] Tim French and Hans van Ditmarsch. Undecidability for arbitrary public announcement logic. In C. Areces and R. Goldblatt, editors, *Advances in Modal Logic*, volume 7, pages 23–42, 2008.
- [9] James Hales. Arbitrary action model logic and action model synthesis. In *2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 253–262, 2013.
- [10] Barteld Kooi and Brian Renne. Arrow update logic. *Review of Symbolic Logic*, 4(4):536–559, 2011.
- [11] Jan Plaza. Logics of public communication. In M.L. Emrich, M.S. Phifer, M. Hadzikadic, and Z.W. Ras, editors, *Proceedings of the Fourth International Symposium on Methodologies for Intelligent Systems, Poster Session Program*, pages 201–216, 1989. Reprinted as [12].
- [12] Jan Plaza. Logics of public communication. *Synthese*, 158:165–179, 2007.
- [13] Hao Wang. Proving theorems by pattern recognition – II. *Bell System Technical Journal*, 40(1):1–41, 1961.